AD-A105 187 YALE UNIV NEW HAVEN CT DEPT OF COMPUTER SCIENCE F/G 12/1
FINAL REPORT FOR CONTRACT F49620-77-C-0037.(U)
AUG 81 M H SCHULTZ F49620-77-C-0037
UNCLASSIFIED AFOSR-TR-81-0656 NL

1 OF 1
AD
A105187

END
DATE
FILMED
10 81
DTIC

Final Report for Contract F49620-77-C-0037

## 1. Direct Methods

Work on direct sparse matrix methods continued under the grant. One of the chief thrusts of our research has been how to use these sparse matrix techniques in situations where primary memory is smaller than problem size.

Along with Andy Sherman of the Department of Computer Science at the University of Texas, we investigated what are called Minimal Storage Methods. Rather than save the factorization in auxiliary storage, we throw away most nonzero entries and recompute them as necessary during back-solution [7, 8]. Surprisingly, for model problems, the work required is less than twice that for conventional sparse elimination, although the bookkeeping overhead does increase somewhat.

We investigated the use of secondary storage in conjunction with band elimination [16, 11]. This work focused on trying to understand and parameterize the general issues involved, designing and analyzing classes of algorithms that use secondary storage, implementing and benchmarking these algorithms, and studying new computer architectures and software systems that would allow us to use secondary storage more effectively to solve banded linear systems.

For sparse elimination, the straightforward approach to auxiliary storage (forming the rows of the factorization one at a time while keeping the previously computed rows in auxiliary storage and fetching them as needed) is grossly inefficient: the I/O overwhelms the computation. It appears, however, that the minimal-storage approach to sparse elimination [8] can be adapted to auxiliary storage and will result in an efficient algorithm for solving very large sparse systems of linear equations.

Our research on iterative methods centered on multi-grid iterative methods and on preconditioned conjugate gradient and conjugate residual methods [4, 3]. While the multi-grid algorithm has been shown mathematically to be asymptotically optimal [2] and it well known that preconditioned conjugate gradient is not [4], recent empirical computer studies [5] indicate that, for problems of practical size, the preconditioned conjugate gradient method is surprisingly competitive. Moreover, very recently Eisenstat [12] showed how to significantly speed up preconditioned conjugate gradient codes based on approximate factorizations, making preconditioned conjugate gradient methods even more competitive.

We investigated extensions of many of the ideas of preconditioned conjugate gradient methods to the class of nonsymmetric matrices with positive definite symmetric parts. Such matrices arise, for example, in finite-difference approximations to the convection-conduction equation [1]. We obtained a number of startling empirical results [13, 9], but while we have some new theory, we still cannot explain all of the experiments. We obtained the first convergence proof [9] of Orthomin [17], one of the algorithms that appear to be most promising in practice. Much theoretical and experimental work remains to be done in this area. The surface has barely been scratched.

## 2. Mathematical Software

In order to disseminate numerical algorithms to the scientific community, numerical analysts must prepare well-documented, modular, portable mathematical software that implements these algorithms. Otherwise algorithms are either ignored because they seem too complicated to program or mis-implemented, sometimes in grossly inefficient ways. One of the prime objectives in our research has been to implement the ideas we develop.

Our work on mathematical software for solving very large sparse systems of linear equations focused both on direct methods, where the major emphasis was on adapting in-memory techniques to situations with limited memory, and on iterative methods, where the major emphasis was on extending preconditioned conjugate gradient methods to nonsymmetric systems.

Along with Andy Sherman of the Department of Computer Science at the University of Texas, we developed a prototype code for Minimal Storage Sparse Elimination [8]. In tests against our own classic Yale Sparse Matrix Package, it proved to be surprisingly competitive for a simple model problem. The same ideas used to implement minimal storage sparse elimination seem to apply to adapting general sparse elimination to auxiliary storage (like disks) in such a way as both to minimize I/O and to maximize the overlap of I/O and computation.

The straightforward implementation of sparse elimination [10, 14] does not mesh well with the latest class of super-computer, the vector processor. Vector processors differ from the more conventional scalar processors in their ability to operate on vectors, sequences of contiguous or regularly spaced memory locations, far more efficiently than on the components individually. (Thus the time to add together two vectors of length n would be s+tn, where s denotes the startup time and t (<< s) the time per addition, whereas the time to add together two scalars would be s+t.) To take advantage of this vector hardware, however, it is necessary to "vectorize" the algorithms used, sometimes replacing a nonvectorizable one that would run faster on a scalar machine with a slower but vectorizable one. Unfortunately, the innermost loop in sparse elimination, where the bulk of the computation is done, is of the form

```
      DO 1 J=JMIN,JMAX
    1   ROW(JU(J)) = ROW(JU(J)) + UKI*U(J)
```

which involves a scatter-fetch (creating a contiguous vector from randomly scattered memory locations), adding one multiple of a vector to another, and then a scatter-store. Only the second phase is vectorizable. On the other hand, the MSSE approach to sparse elimination does appear to vectorize well and could run reasonably fast.

We have investigated a number of variants of the multi-grid approach for solving finite-difference approximations to linear boundary-value problems for elliptic partial differential equations. To do uniform comparisons, we have developed a package implementing multi-grid in a fairly general manner [6].

We investigated extensions of many of the ideas underlying preconditioned conjugate gradient methods to the class of nonsymmetric matrices with positive definite symmetric part. In order to compare the different iterative methods and preconditionings in a common environment, we created a prototype package that implements these methods [13], the user interface being similar to that used in ITPACK [15]. As we gain more experience about which methods are most effective, we hope to refine this prototype into mathematical software.

## REFERENCES

[1] E. Allgower and K. Georg.
Simplicial and continuation methods for approximating fixed points and solutions to systems of equations.
SIAM Review 22(1):28-85, 1980.

[2] R. E. Bank and T. Dupont.
An Optimal order process for solving elliptic finite element equations.
University of Chicago, 1978.

[3] R. Chandra, S. C. Eisenstat, and M. H. Schultz.
The modified conjugate residual method for partial differential equations.
In Proc. Second IMACS Inter. Symp., pages 13-19. IMACS, Lehigh University, 1977.

[4] Chandra, R.
Conjugate Gradient Methods for Partial Differential Equations.
PhD Thesis, Yale University, 1978.

[5] Dendy, J.E.m and Hyman, J.M.
Multi-grid and ICCG for problems with interfaces.
In M.H. Schultz, editor, Elliptic Problem Solvers. Academic Press, New York, 1981.

[6] Douglas, C.
A multi-grid package for solving elliptic boundary value problems.
Manuscript, Department of Computer Science, Yale University.

[7] Eisenstat, S.C., Schultz, M.H. and Sherman, A.H.
Minimal storage band elimination.
In D. J. Kuck, D. H. Lawrie, and A. H. Sameh, editors, Proceedings of the Symposium on High Speed Computer and Algorithm Organization, pages 273-286. Academic Press, New York, 1977.

[8] Eisenstat, S.C., Schultz, M.H. and Sherman, A.H.
Software for sparse Gaussian elimination with limited core storage.
In Proceedings of the Sparse Matrix Meeting, pages 135-153. SIAM, Philadelphia, 1978.

[9] S. C. Eisenstat, H. Elman and M. H. Schultz.
Variational iterative methods for nonsymmetric systems of linear equations.

[10] S. C. Eisenstat, M. C. Gursky, M. H. Schultz, and A. H. Sherman.
Yale sparse matrix package I: The symmetric codes.
Technical Report 112, Department of Computer Science, Yale University, 1977.

[11] S. C. Eisenstat, J. R. Perry, and M. H. Schultz.
Some observations on auxiliary storage band solvers.
ACM-SIGNUM Mathematical Software Conference, Albuquerque, 1977.

[12] S. C. Eisenstat.
Efficient implementation of a class of preconditioned conjugate gradient methods.
Technical Report 185, Department of Computer Science, Yale University, 1980.

[13] H. Elman.
Variational Iterative Methods for Solving Nonsymmetric Linear Systems of Equations.
PhD Thesis, Yale University, 1981.

[14] A. George and J. W. H. Liu.
User Guide for SPARSPAK: Waterloo Sparse Linear Equations Package.
Research Report CS-78-30, Department of Computer Science, University of Waterloo, 1978.

[15] R. G. Grimes, D. R. Kincaid, W. I. MacGregor, and D. M. Young.
Itpack report: Adaptive iterative algorithms using symmetric sparse storage.
Technical Report CNA-139, *Center for Numerical Analysis*, The University of Texas at Austin, 1978.

[16] Perry, J.R.
Secondary Storage Methods for Solving Banded Linear Systems.
PhD Thesis, Yale University, 1981.

[17] P. K. W. Vinsome.
Orthomin, an iterative method for solving sparse sets of simultaneous linear equations.
Paper SPE 5729, Fourth Symposium of Numerical Simulation of Reservoir Performance of the SPE., Los Angeles, California, 1976 .

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

## REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

| | | |
|---|---|---|
| 1. REPORT NUMBER<br>AFOSR-TR- 81 -0656 | 2. GOVT ACCESSION NO.<br>AD | 3. RECIPIENT'S CATALOG NUMBER |
| 4. TITLE (and Subtitle)<br>FINAL REPORT FOR Contract F49620-77-C-0037 | 9 | 5. ~~TYPE OF~~ REPORT & PERIOD COVERED<br>Rept for ~~Final~~ 1 JAN 77 — 31 DEC 80 |
| | 18 | 6. ~~PERFORMING ORG. REPORT NUMBER~~ |
| 7. ~~AUTHOR(s)~~<br>Martin H. Schultz | AFOSR<br>TR-81-0656 | 8. CONTRACT ~~OR GRANT~~ NUMBER(s)<br>15 F49620-77-C-0037 |
| 9. PERFORMING ORGANIZATION NAME ~~AND ADDRESS~~<br>Department of Computer Science<br>Yale University<br>New Haven CT 06520 | | 10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS<br>61102F, 2304/A3 17 A3 |
| 11. CONTROLLING OFFICE NAME AND ADDRESS<br>Air Force Office of Scientific Research/NM<br>Bolling AFB DC 20332 | 11 | 12. REPORT DATE<br>AUG 81 |
| | | 13. ~~NUMBER OF~~ PAGES<br>6 12 7 |
| 14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) | | 15. SECURITY CLASS. (of this report)<br>UNCLASSIFIED |
| | | 15a. DECLASSIFICATION DOWNGRADING SCHEDULE |

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The investigators associated with this contract produced 17 documents on the broad area of sparse matrices. They recorded research on algorithmic and software development of direct sparse linear equation solvers. Also, the examination of problems involving sparse matrices that arise in the numerical solution of partial differential equations is included. The development of the Yale Sparse Matrix Package, a computer software package to efficiently solve directly large sparse systems, is indeed on of the highlights of this research. It has been independently tested and found exceptional in solving symmetric problems.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

407051

JOB

ATE

LMED

— 8